

Raspberry Pi / ROS-basiertes Low-Cost-Multikamerasystem zum mobilen Einsatz im Feldversuchswesen

Low-Cost Multicamera System based on Raspberry Pi / ROS for mobile use in field trials

Matthias Igelbrink, Axel Höh, Andreas Linz und Arno Ruckelshausen

Hochschule Osnabrück, Competence Center of Applied Agricultural Engineering (COALA), Albrechtstr. 30, 49076 Osnabrück,

E-Mail: matthias.igelbrink@hs-osnabrueck.de / a.ruckelshausen@hs-osnabrueck.de

Zusammenfassung:

Die Sensor- und Datenfusion spielt in landwirtschaftlichen Anwendungen eine zunehmend bedeutendere Rolle, sowohl für neue Applikationen als auch die Verbesserung der Robustheit der Sensorik unter dem Einfluss variabler Störgrößen. Dies betrifft insbesondere das Feldversuchswesen zur Evaluation von Sensoren zur Messung von Pflanzen-, Boden- oder Umgebungseigenschaften, ein Beispiel ist die Bestimmung von Pflanzenparametern in der Pflanzenzüchtung (Phänotypisierung). Die Anschaffungskosten für diese Sensorsysteme sind in der Regel hoch. Zudem ist man häufig an proprietärere Software gebunden, welche zu einer Abhängigkeit von Firmen oder Lizenzgebühren führt. Hinzu kommt, dass sich die Synchronisierung der Sensordaten sehr schwierig gestaltet. Das hier vorgestellte mobile Multikamerasystem besteht aus Low-Cost Komponenten, basiert auf freier Software, und ist flexibel erweiterbar. Die eingesetzte Middleware ROS (Robot Operating System) verwaltet die Kommunikation zwischen verschiedenen Kameras und Sensoren und versieht jedes Datenpaket mit einem Zeitstempel. Dies schafft die Voraussetzung für eine Daten- bzw. Sensorfusion und stellt somit eine kostengünstige Lösung für eine erste Evaluation von Sensoren auf dem Feld dar.

Deskriptoren: Low-Cost Imaging, Multikamera-Systeme, Sensorfusion, ROS, Raspberry Pi, Open-Source-Software

Abstract:

Sensor- and data fusion gets more and more important in agriculture applications. This applies not only for new applications, it helps also to improve the robustness of sensors and sensor data in the influence of variable disturbance values. In field trials the evaluation of sensors is necessary for measuring plant-, soil- and environmental parameters. A common example is "phenotyping", where many parameters for the characterisation of plants are collected on the field. Usually, commercial sensor systems are expensive, and

they are mostly shipped with proprietary software, which can generate licence fees. Furthermore, synchronisation between different sensor systems can be a complicated task.

In this paper an easily expandable and portable low-cost multi camera- and sensor system is introduced. The applied middleware ROS (Robot Operating System) manages the communication between all cameras and sensors, and tags every data packages with a timestamp, which is a requirement for the sensor- and data fusion. The open source software and the low-cost components leads to an inexpensive system for first sensor evaluations in field trials.

Keywords: Low-Cost Imaging, Multicamera systems, Sensor fusion, ROS, Rapsberry Pi, Open-source software

1 Einleitung

Die Autoren beschäftigen sich intensiv mit der Thematik der Erfassung von Pflanzenparametern mit Hilfe von Sensoren und Sensorsystemen im Feldversuchswesen (Möller, K. & Ruckelshausen, A., 2015). Der Schwerpunkt liegt auf der nicht invasiven und präzisen Bestimmung von morphologischen und spektralen Eigenschaften von Pflanzen (Busemeyer, L, et al., (2013) / Makeen, K, et al., (2012) und Paulus, S., et al., (2015)). Üblicherweise sind die Anschaffungskosten der hier genutzten Sensorsysteme sehr hoch. Zudem ist man häufig an eine proprietäre Software gebunden, die die Flexibilität einschränkt und unter Umständen noch Folgekosten in Form von Lizenzgebühren erzeugt. Ein weiterer Nachteil tritt auf, wenn man die Sensoren bzw. die Sensordaten fusionieren will. Hier ist eine Synchronisation erforderlich, die wiederum durch unterschiedliche Software oder Betriebssysteme erschwert wird, oder auch nicht realisierbar ist. Um neue Methoden zur Pflanzenparametererfassung im Feldversuchswesen zu evaluieren, und Sensordaten zu fusionieren, stellt das Low-Cost-Multikamerasystem ein schlagkräftiges Hilfsmittel dar.

Angesichts der Weiterentwicklungen in den Bereichen der Hardwaretechnologie existiert heutzutage eine hohe Verfügbarkeit von Sensoren auf dem Markt. Diese Tatsache sorgt dafür, dass z.B. die Preise für Rechner-/Kamera-Systeme keine wesentliche Rolle mehr spielen. Zudem ermöglichen Fortschritte im Bereich der Open-Source Software eine flexible und unabhängige Lösung für eine Entwicklung von kostengünstigen Sensorsystemen. Im Folgenden wird ein Low-Cost-Multikamerasystem vorgestellt, das mit einer Vielzahl von Kamerasystemen (RGB, NIR, Wärmebild, 3D, etc.) und weiteren Sensoren (GPS, Beschleunigungssensoren, Temperatur, Feuchte, etc.) parallel bestückt werden kann. Basierend auf dem Open Source Framework ROS sind die Komponenten auf einer Kommunikationsebene vernetzt, die optional alle Datenpakete mit einem Zeitstempel versieht, und somit eine Fusion der Daten ermöglicht. Ein erstes System wurde mit 2 RGB-, einer NIR-, einer Wärmebildkamera, und einem GPS-Modul ausgestattet. Die effektiven Kosten liegen hier bei ca. 700€.

2 Multikamerasystem

Verwendete Hardware – Rechner

Die Anforderungen an das Kamerasystem sind neben einer flexiblen Erweiterbarkeit, und Low-Cost auch eine umfassende Softwareunterstützung. Diese Kriterien konnten mit dem Raspberry Pi 3 Model B in Verbindung mit dem Betriebssystem Raspian Jessie erfüllt werden. Der Raspberry Pi verfügt über mehrere Kommunikationsschnittstellen (USB, Bluetooth, WLAN, Ethernet). Hinzu kommen 40 GPIO-Pins für die Einbindung zusätzlicher Sensorik über SPI, I2C oder UART. Für eine spezielle Raspberry-Kamera gibt es eine eigene Schnittstelle (CSI). Da der Raspberry für den schulischen Bereich entwickelt wurde, ist auch das Low-Cost-Kriterium erfüllt. An Software steht kostenlos ein Betriebssystem (Raspian, eine Variante von Debian-Linux) mit all seinen Open-Source-Erweiterungen und –Programmpaketen zur Verfügung. Mit OpenCV, einer weit verbreiteten Bildbearbeitungsbibliothek können die Kameradaten bereits im Multikamerasystem vorverarbeitet werden. Das Framework ROS (Robot Operating System) ist ebenfalls auf dem Raspberry nutzbar und erlaubt einen modularen Systemaufbau und größtmögliche Skalierbarkeit ebenso wie einen einfachen Anschluss an externe PC-Systeme zur Datenauswertung.

Verwendete Hardware - Sensoren

Das Multikamerasystem wurde als Beispiellösung mit vier Kameras realisiert, kann aber nahezu beliebig erweitert werden. Zwei der vier eingebauten Kameras sind RGB-Kameras. Diese können aufgrund ihrer Anordnung auch als Stereokamerasystem für 3D-Imaging verwendet werden. Eine weitere RGB-Kamera besitzt kein NIR-Sperrfilter, kann also nahes Infrarotlicht detektieren und deshalb in Kombination mit einer der beiden anderen RGB-Kameras Informationen über Vegetationsindizes liefern. Die verwendeten RGB-/NIR-Kameras wurden speziell für den Raspberry Pi entwickelt und besitzen einen 5 Megapixel CMOS-Sensor mit einer max. Auflösung von 2592 x 1944.

Das Kameramodul „Lepton“ von der Firma FLIR detektiert Infrarotstrahlung im Wellenlängenbereich 8-14 μm . Es liefert Wärmebilder mit einer Auflösung von 80x60 Pixel. Die Temperaturauflösung beträgt dabei 50mK.

Zusätzlich zu den Kameras ist ein GPS-Modul integriert, um Informationen über Ortskoordinaten erhalten zu können. Der Einbau weiterer Sensoren ist aufgrund der vielfältigen Anschlussmöglichkeiten der verwendeten Raspberry Pis ebenfalls möglich.

Systemaufbau

Der Hardwareaufbau (Bild 1) des Multi-Kamerasystems besteht in der Beispiellösung aus vier Raspberry Pis. Davon werden drei Pis als Clients für die Bereitstellung von Sensordaten bzw. als Auswerteeinheit der Kameras verwendet. Der Anschluss der Raspberry-

Kameras erfolgt mittels 15-poligem Flachbandkabel an das Camera Serial Interface (CSI) direkt auf die Platine des Raspberry Pis. Die Wärmebildkamera sowie eventuelle weitere Sensoren werden über die GPIO-Ports angebunden. Der vierte Raspberry Pi dient als Steuereinheit sowie als Server für Ros. Auch das Speichermedium für die aufgenommenen Daten (ein USB-Stick) sowie das GPS-Modul ist hier angeschlossen. Zur einfachen Bedienung des Kamerasystems dient ein Touch-Display am Display Serial Interface (DSI). Für die Datenübertragung zwischen den vier Raspberrys sowie als Interface zu externen PCs wird ein 100 Mbit Ethernet-Switch verwendet.

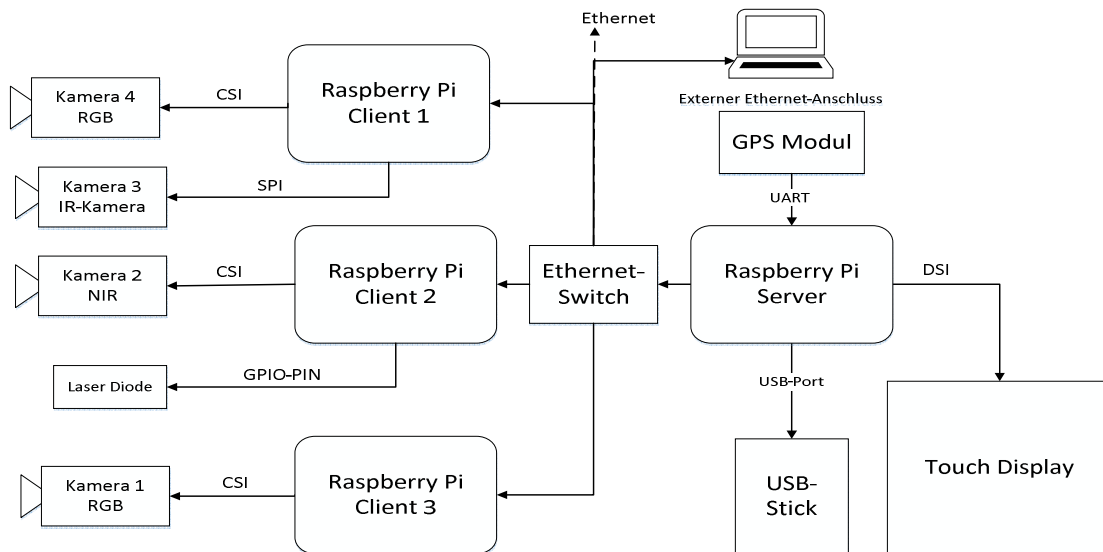


Bild 1: Hardwareaufbau

Verwendete Software

Für die selbst erstellte Systemsoftware des Multikamerasystems wurde Python 2.7 verwendet. Python ist eine recht übersichtliche Programmiersprache, die aber durch Einbindung zusätzlicher Bibliotheken (OpenCV, ROS, Hardwaretreiber) erweiterbar und dadurch sehr flexibel ist.

Die interne Kommunikation der Raspberry Pis untereinander sowie die Übertragung von Sensordaten wird über das Robot Operating System (ROS, Version „Indigo“) realisiert. Dabei handelt es sich um ein Framework, das ursprünglich für Roboteranwendungen gedacht war und Kommunikation und Datenaustausch zwischen verteilten Prozessen (Nodes) ermöglicht (Quigley, M. et al 2009). ROS verfügt dafür über zwei Kommunikationsverfahren: eine Service-orientierte, synchrone Kommunikation (Peer2Peer) sowie eine asynchrone Kommunikation für die Übertragung von Sensordaten. Die asynchrone Kommunikation baut auf dem Publisher-Subscriber-Modell auf. Dieses erlaubt die Bereitstellung von Daten eines Publishers über definierte virtuelle Kommunikationsbusse, auf denen sich ein oder mehrere Prozesse (Subscriber) einschreiben können. Alle Daten sind mit einem Zeitstempel versehen, was eine Synchronisierung der Sensordaten ermöglicht und eine spätere Analyse vereinfacht.

Für die Vorverarbeitung der Bilddaten auf den Raspberry Pis wird OpenCV 2.4.11 verwendet. Diese Bildbearbeitungsbibliothek lässt sich mit ROS und Python kombinieren. Sie zeichnet sich durch eine Vielzahl von Algorithmen sowie eine stetige Weiterentwicklung unter Open-Source-Lizenz aus.

Um eine Bedienung über das Touch-Display zu gewährleisten, wurde eine grafische Benutzeroberfläche programmiert. Dafür wurde die Pythonbibliothek PyQt 4 (Python API für die C++ Klassenbibliothek Qt) verwendet. Qt bietet nicht nur umfangreiche Funktionen für die Programmierung grafischer Oberflächen an, sondern auch eine problemlos Einbindung in das Robot Operating System.

ROS-Netzwerk des Multikamerasystems

Die Kommunikation im ROS-Netzwerk findet zwischen Nodes (Prozessen) statt (Bild 2). Die Verbindungsdaten der einzelnen Nodes werden über den sogenannten ROS-Master verwaltet. Dabei kann dieser eine beliebige Anzahl an Nodes verwalten. Der Datenaustausch erfolgt allerdings direkt zwischen den Prozessen über Topics (virtuelle Kommunikationsbusse). Im Multikamerasystem bildet jeder Raspberry Pi einen eigenen Node, der jeweils über einen eigenen Service für die Ansteuerung sowie einen definierten Kommunikationsbus für die Bereitstellung von Sensordaten verfügt (Bild 2). Dieses Konzept erlaubt einen modularen Aufbau des Systems und ermöglicht eine nahezu beliebige Erweiterung durch zusätzliche Rechner-/Kamerasysteme.

Realisierung des Multikamerasystems

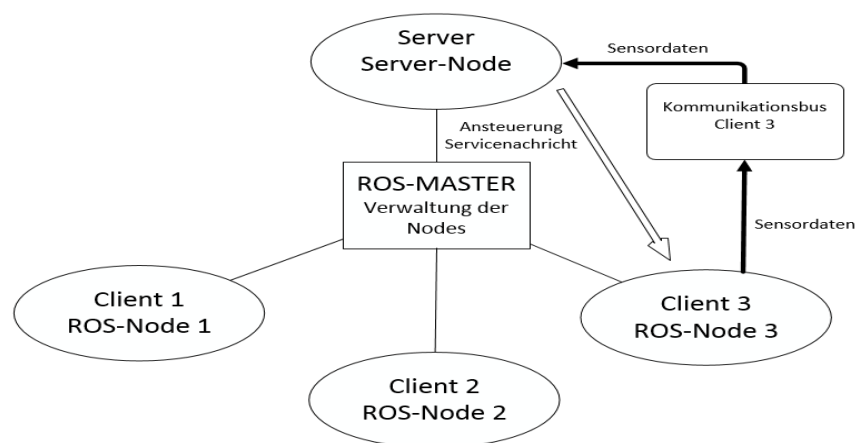


Bild 2: ROS-Netzwerk und Ansteuerung

Das Multikamerasystem wurde kompakt in einem gemeinsamen Gehäuse eingebaut (Bild 6) und ist, unter der Voraussetzung einer ausreichenden Spannungsversorgung, mobil einsetzbar. Das System kann über die grafische Benutzeroberfläche (Bild 6.2) in

Verbindung mit dem Touch-Display leicht bedient werden (Bild 6.1). Durch den modularen Aufbau (Bild 6) kann das System ohne großen Aufwand durch weitere Kameras oder zusätzliche Sensorik erweitert werden. Die aktuell implementierten Funktionen des Kamerasystems umfassen Einzelbild-, Vierfachbild-, Mehrfachbild- und Videoaufnahmen (Bild 6.2). Zudem lässt sich die Bildauflösung frei wählen. Die Framerate variiert je nach eingestellter Auflösung; bei 640x480 lassen sich bis zu 20 fps verarbeiten. Über eine implementierte Kameravorschau lassen sich die Bilder auch live anzeigen (Bild 6.1).

Das GPS-Modul kann über das Menu ein- und ausgeschaltet werden. Die GPS-Daten werden über das ROS-Netzwerk bereitgestellt und stehen somit auch externen Systemen zur Verfügung.

Die Sensordaten lassen für eine spätere Analyse auf der SD-Karte oder einem USB-Stick abspeichern. Des Weiteren kann das System über eine externe Ethernet-Schnittstelle angesteuert und ausgelesen werden.

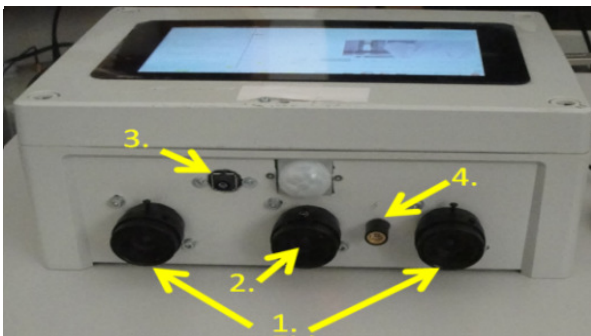


Bild 3: Vorderansicht Multikamerasystem, RGB-Kameras (1), RGB+NIR-Kamera (2), Wärmebildkamera (3), Laserdiode (4)

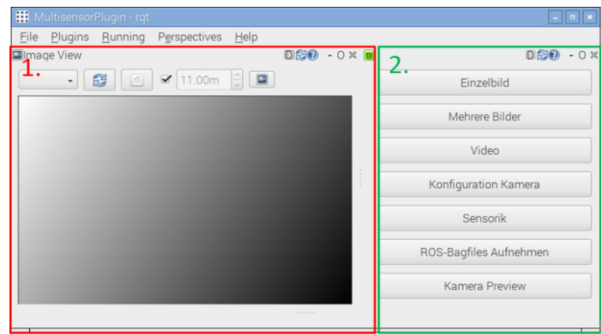


Bild 4: Benutzeroberfläche Multikamerasystem, Kameravorschau (1), Auswahlmenu (2)



Bild 5: GPS-Modul (1), Touch-Display (2), USB-Stick (3), externe GPS-Antenne (4)

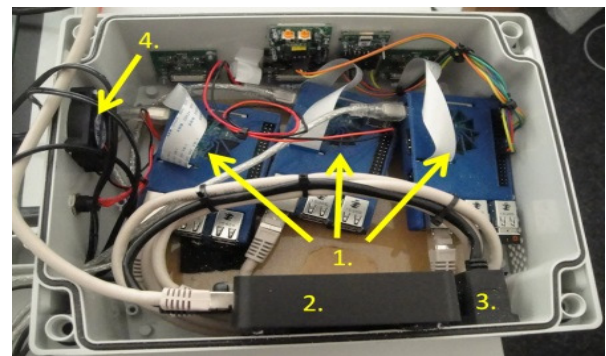


Bild 6: Innenansicht Multikamerasystem, Raspberry Pi (1), Ethernet-Switch (2), externe Ethernet-Schnittstelle (3), Lüfter (4)

3 Anwendungsbeispiel

Ziel der Anwendung des Systems ist eine erste Evaluation von (überwiegend) bildgebenden Sensoren und Sensorfusionsansätzen für die darauf aufbauende Entwicklung von kostengünstiger Multi-Sensorsystemen und Algorithmen im Feldversuchswesen (oder

anderen Anwendungsgebieten). Dabei können Kamerabilder mit verschiedenen Signaturen, weitere Sensordaten sowie zusätzliche (externe) Daten verwendet werden.

Im nachfolgenden Beispiel wird als klassisches Anwendungsbeispiel die Benutzung von RGB- und NIR-Kamera für die Erstellung eines NDVI-Bildes vorgestellt. Dafür wurden Pflanzen im Außen- und Laborbereich aufgenommen.

Mithilfe der NDVI (Normalized Differenced Vegetation Index)-Aufnahmen lassen sich Informationen über die Vitalität von Pflanzen gewinnen. Diese Informationen beruhen auf der Tatsache, dass gesunde Pflanzen im sichtbaren Spektralbereich relativ wenig und im nachfolgenden nahen Infrarotbereich relativ viel Strahlung reflektiert. Dieser Effekt korreliert mit dem vorhandenen Anteil an Chlorophyll in den Pflanzen, was Rückschlüsse auf die photosynthetische Aktivität, Vitalität sowie die Dichte der Vegetationsdecke zulässt.

Die verwendete NIR-Kamera unterscheidet sich von den RGB-Kameras dadurch, dass kein IR-Sperrlichtfilter auf dem CMOS-Sensor angebracht ist. Da sich das übliche Bayer-Pattern trotzdem auf dem Sensor befindet, erhält man statt der drei Farbkanäle „rot“, „grün“, „blau“ nun die Farbkanäle „rot + NIR“, „grün“ und „blau“. Die Kamera detektiert damit einen Wellenlängenbereich von 380nm bis ca. 1000nm (Bild 7.2), während die RGB-Kamera Wellenlängen im sichtbaren Bereich von 380nm bis 750nm erfasst (Bild 7.1). Die für die Berechnung benötigten spektralen Informationen im sichtbaren bzw. nahen Infrarotbereich lassen sich mit einer Aufnahme des Multikamerasystems gewinnen. Im Anschluss daran werden beide Bildaufnahmen mit OpenCV in die einzelnen Farbkanäle zerlegt und nur die Farbkanäle „rot“ bzw. „rot + NIR“ verwendet. Da die Information „NIR“ nicht separat vorliegt, wird die Gleichung (1) etwas abgewandelt zu Gleichung (2), was aber natürlich den gleichen Wert ergibt:

$$NDVI = \frac{NIR - Rot}{NIR + Rot} \quad (\text{Gleichung 1})$$

$$NDVI = \frac{(Rot + NIR) - 2x(Rot)}{Rot + NIR} \quad (\text{Gleichung 2})$$

Das daraus gewonnene NDVI-Bild (Bild 7.3) wurde für eine übersichtlichere Darstellung mittels OpenCV noch in ein Falschfarbenbild umgewandelt (Bild 7.4, Bild 8.4). Dabei gibt rot einen hohen NDVI an, während blaue Bereiche einen niedrigen NDVI aufweisen.

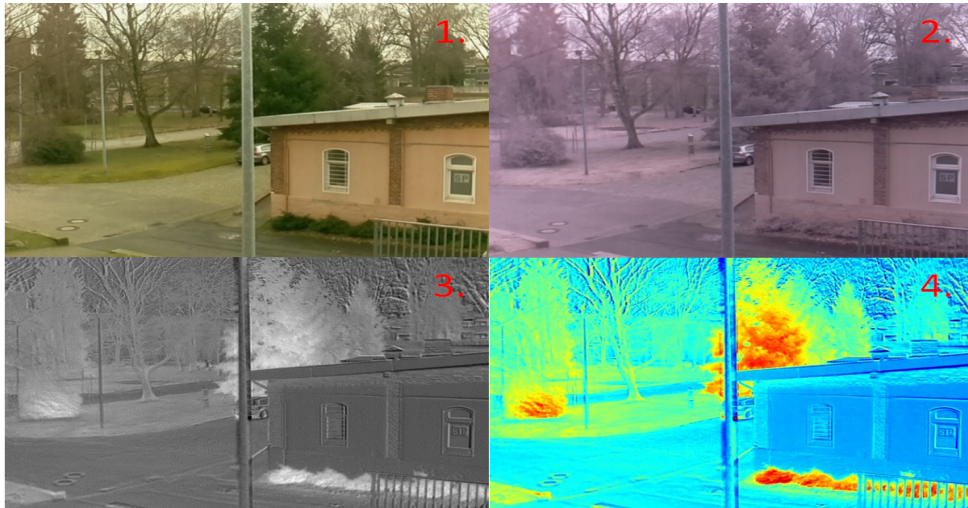


Bild 8: Außenaufnahme, RGB-Bild (1), NIR-Bild (2), NDVI (3), Falschfarbenbild NDVI (4)

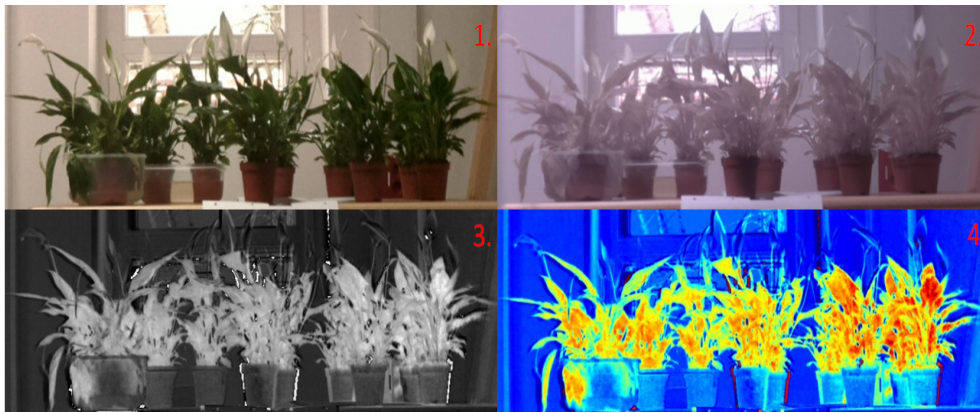


Bild 7: Pflanzen im Labor, RGB-Bild (1), NIR-Bild (2), NDVI (3), Falschfarbenbild NDVI (4)

4 Ausblick

Obiges Beispiel zeigt eine erste, im Pflanzenbau übliche Anwendung. Da die Anzahl der eingebunden Sensoren nahezu unbegrenzt ist, sind weitere Szenarien, bzw. das mittloggen zusätzlicher Information möglich. Denkbare Sensorinformationen wären Temperatur, Druck, Feuchte, Helligkeit, etc. Das eingebaute GPS-Modul bietet die Option die erfassten Daten direkt mit den Orts- und Zeitkoordinaten zu verknüpfen. Des Weiteren kann der eingebaute Kreuzlinienlaser für eine optische Markierung zu Kalibrierungszwecken oder für eine Entfernungsmessung durch Triangulation genutzt werden. Da alle gespeicherten Datenpakete mit Zeitstempeln versehen sind, können bei einer späteren Auswertung die Bilder der beiden RGB-Kameras synchronisiert und zur Berechnung eines Tiefenbildes aus Stereobildern verwendet werden. Dies ist möglich, da der Abstand zwischen beiden Kameras bekannt ist. Aus pflanzenbaulicher Sicht ist auch die bildgebende Temperatur der Blattoberfläche oder des Bodens interessant. Diese Informationen werden durch das FLIR-Modul zur Verfügung gestellt.

Literaturverzeichnis

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: (2009). **ROS: an open-source Robot Operating System.** Journal ICRA workshop on Open-Source Software, Internet: <http://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf>

Busemeyer, L.; Mentrup, D.; Möller, K.; Wunder, E.; Alheit, K.; Hahn, V.; Maurer, H.P.; Reif, J.C.; Würschum, T.; Müller, J.; Rahe, F.; Ruckelshausen, A. ; BreedVision — A Multi-Sensor Platform for Non-Destructive Field-Based Phenotyping in Plant Breeding, Sensors, 2013, Vol. 13, pp. 2830-2847.

Möller, K., Ruckelshausen, A.: "Datenmanagement für Ultra-High-Precision-Phenotyping in Feldversuchen", in: **Ruckelshausen, A., Schwarz, H.-P., Theuvsen, B.** (Hsrg.): Informatik in der Land-, Forst-, und Ernährungswirtschaft, Referate der 35. GIL-Jahrestagung, 23.-24. Februar 2015, Geisenheim, S. 117-120, ISBN 978-3-88579-632-9.

Makeen, K., Kerksen, S., Mentrup, D., Oelmann, B., Ruckelshausen, A.: Multiple Reflection Ultrasonic Sensor System for Morphological Plant Parameters, Bornimer Agrartechnische Berichte, Heft 78, 2012, S. 110-116.

Paulus, S., Dornbusch, T., Nagel K.: Introducing multi-sensor platforms for detailed characterization of plant canopies during the entire growing season, 2015, ATB Workshop Potsdam, Internet: http://www.lemnatec.com/fileadmin/user_upload/PDFpool/LemnaTec_article_CBA_UAS_Paulus.pdf

Strothmann, W., Tsukor, V., Ruckelshausen, A.: "In-Field-Labeling-HMI für automatische Klassifizierung bei der Pflanzen- und Erntegutcharakterisierung mittels bildgebender Sensordaten", in: Ruckelshausen, A., Schwarz, H.-P., Theuvsen, B. (Hsrg.): Informatik in der Land-, Forst-, und Ernährungswirtschaft, Referate der 35. GIL-Jahrestagung, 23.-24. Februar 2015, Geisenheim, S. 177-180, ISBN 978-3-88579-632-9.

Meltebrink, C.; Linz, A.; Ruckelshausen, A.: "ROS-basiertes Validierungskonzept für Sicherheitskonzepte von autonomen Agrarrobotern", Referate der 36. GIL-Jahrestagung, 22.-23. Februar 2016, Osnabrück, S. 121-124, ISBN 978-3-88579-647-3.

Bangert, W., Kielhorn, A., Rahe, F., Albert, A., Biber, P., Grzonka, S., Haug, S., Michaels, A., Mentrup, D., Hänsel, M., Kinski, D., Möller, K., Ruckelshausen, A., Scholz, C., Sellmann, F., Strothmann, W., Trautz, D.: Field-Robot-Based Agriculture: "RemoteFarming.1" and "BoniRob-Apps", 71. Tagung LAND. TECHNIK - AgEng 2013, pp. 439-446, ISBN 978-3-18-092193-8.